

Appendix G

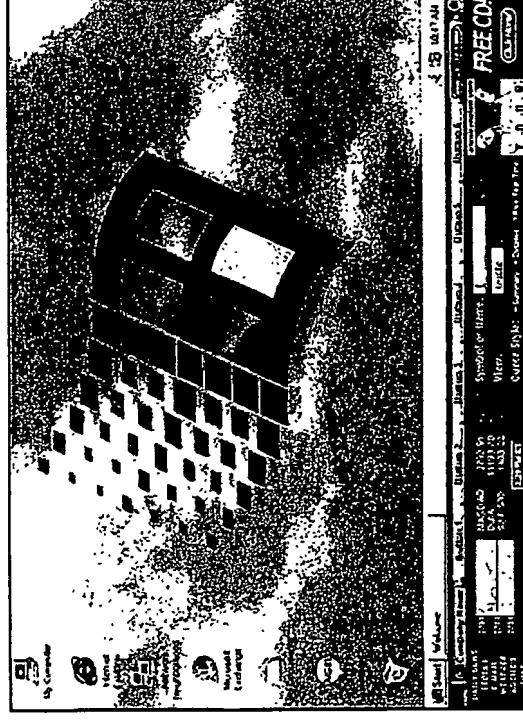
[illegible]

xSides

xSides Overview

xSides provides a customizable, multi-task application that resides outside the confines of the Windows™ desktop.

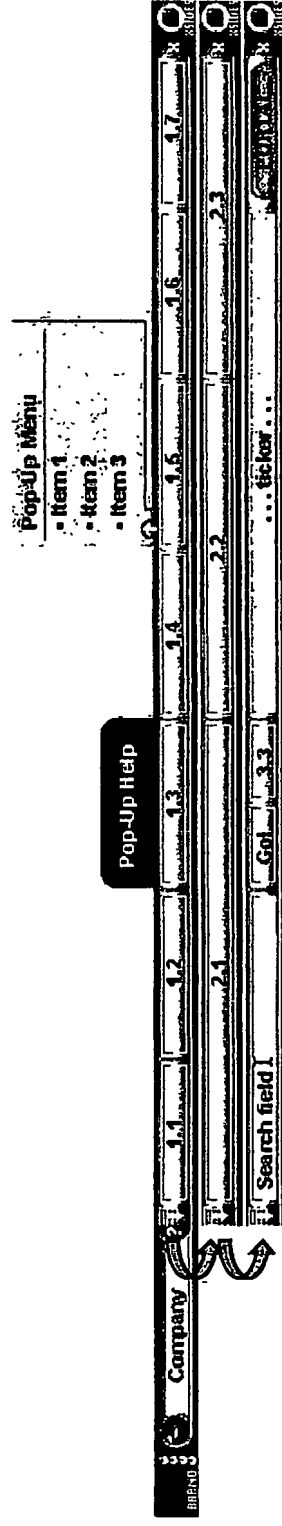
- Buttons, logos, and menus can be customized to create unique sides, using the Level 1 and Level 2 Application Programming Interface (API)
- A Custom side allows for limited customization of special buttons by the user



xSides Side Applications

xSides rotates to allow many sides on one application.

- A side is one row of objects, such as buttons and tickers.
- Sides can be accessed through the rotator buttons or through a navigation menu.
- Objects of one side may be designed to work together as a Web browser, while the objects of another side may be a row of buttons linked to various pages of a company's Web site.



xSides Customization

Buttons, logos, and menus can be customized by the developer to create unique sides using the Level 1 and Level 2 Application Programming Interface (API).

Level 1 API:

- Create new sides by assigning names and paths to standard buttons in the DAT files.
- Standard buttons include clickable buttons, tickers, spacers, and rotators of various widths denoted by their BMP ID#.
- Assign names and paths to cascading menus in a DAT file.
- The content of a DAT file is in XML.

xSides Customization

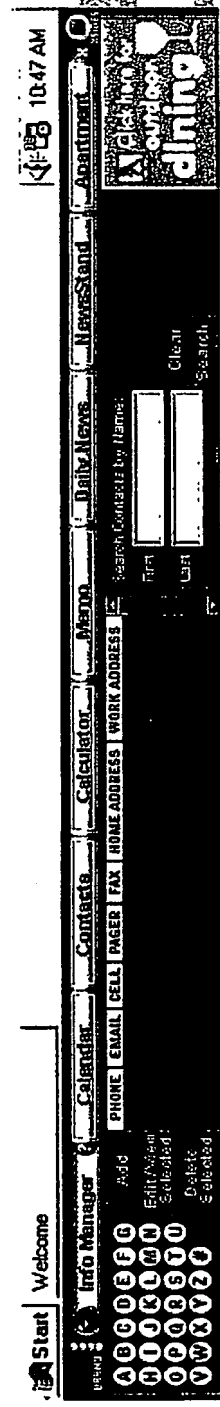
Level 2 API:

1. Add functionality to objects, like buttons and tickers, through a set of callable DLL functions and messages.
2. DLL support includes side object resources like bitmaps, palette, and dialogs.
3. Requires a standard Windows development environment

Portal Overview

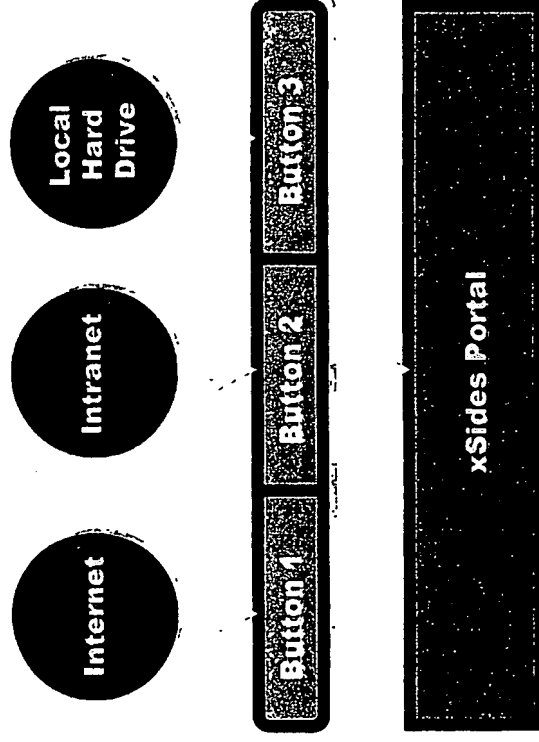
The Portal area functions exclusively with xSides. The xSides Portal launches underneath a side and responds to the xSides commands. Like xSides, the xSides Portal cannot be covered by Windows.

- xSides Portal source can come from the Internet, intranet or locally from the hard disk.
- Portal creation is as simple as creating a Web page.
- Portal height is adjustable.



Portals' Source

The xSides Portal source can come from the Internet, intranet, or locally, from the hard disk. Multiple portals can be assigned to a single product, provided each portal has its own link through a button or menu option. Only one portal can be displayed at a time.



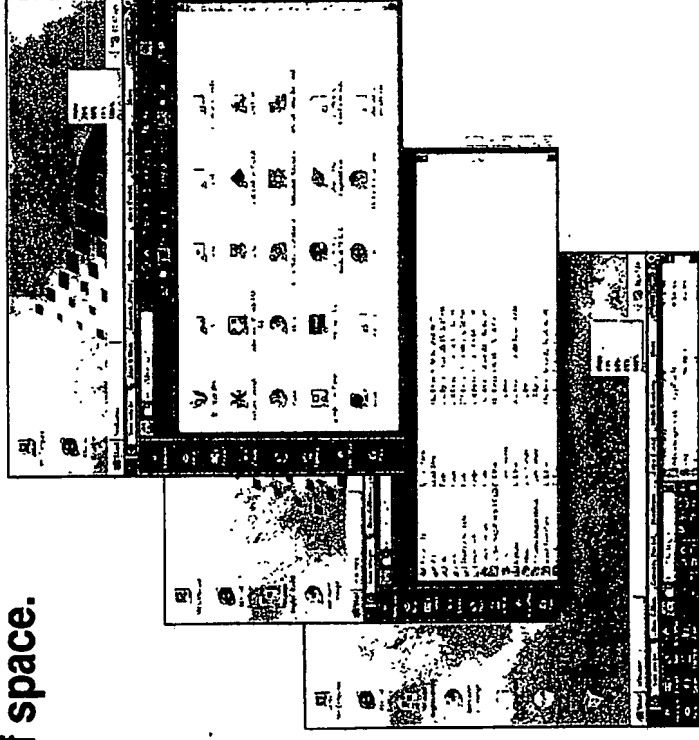
Portal Development Standards

- Server-side data manipulation can be done with any languages or programs as long as you follow Internet Explorer 4.0 standards for HTML, Dynamic HTML, JavaScript and Cascading Style Sheets for client-side output.
- If features are developed using Internet Explorer 5.0 standards, a backward-compatible solution should be in place.
- Cursors will not appear in text and text area elements.
- Some form elements cannot function in a xSides Portal area, but a JavaScript equivalent is available on request.
- Applications like Java applets or Macromedia's Flash are not currently supported.

Portal Adjusting Height

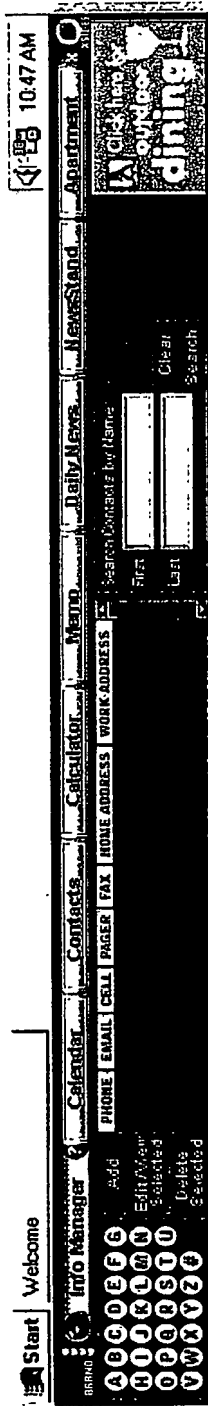
The xSides Portal height can be adjusted for maximum control and efficiency of space.

- Adjust the height in pixels by changing a single value in an INI file.
- Normal xSides Portal height is 70 pixels.
- Portal height can range from 0 (no portal) to maximum vertical screen resolution minus the height of xSides.



Portal Web-Based Applications

In addition to hosting Web pages, the portal can support applications that run entirely over the Internet, are always up to date, and conserve local hard disk space. The xSides Portal is the perfect environment for running Web-Based Applications.



Portal Web-Based Applications Example: Personal Information Manager (PIM)

The PIM is a time-saving desktop organizer opened from xSides, that functions inside the portal. With the xSides User Registration feature, a user's personal PIM data can be accessed from any computer.

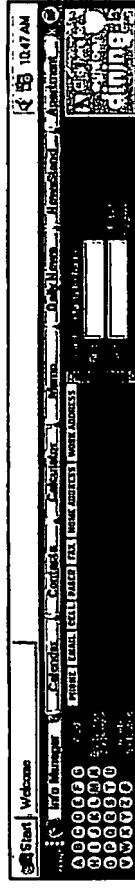
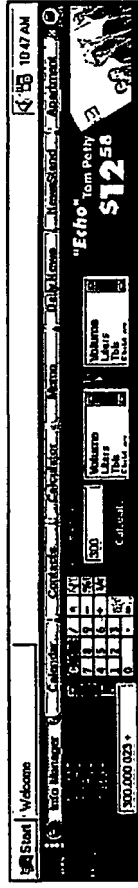
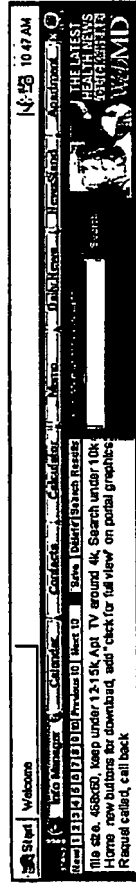
The screenshot shows a web-based Personal Information Manager (PIM) interface. At the top, there is a navigation bar with links: Start, Welcome, Info Manager, Calendar, Contacts, Calculator, Metro, Daily News, NewsStand, and Apartment. Below the navigation bar, there is a search section with a label 'Search Contacts by Name:' and two input fields for 'First' and 'Last' names, along with 'Clear' and 'Search' buttons. The main content area displays a list of contacts with columns for Name, Address, Phone, Email, Cell, Pager, Fax, Home Address, and Work Address. A sidebar on the right contains a list of letters A-Z and a search bar.

Portal Web-Based Applications

Example: Personal Information Manager (PIM)

PIM features include:

- Address Book
- Calendar
- Calculator
- Memo Feature
- To-Do List
- Daily News
- Newsstand
- Import/Export Feature



User Registration Overview

User Registration is a client/server application that gathers user information for data processing and storage.

- Allows multiple users
- Access your bar, address book, and calendar from multiple computers
- Access to additional features, promotions, and updates
- Easy to use
- Is a basis for Universal Registration

User Registration How Information is Gathered

- HTML forms gather and submit user information.
- Information is submitted to a server for authentication and storage.

User Registration Functionality

- Registration
- Login
- Logout
- Change Profile
- Forgot My Password

AllSides Overview

- Allows sides to be added and removed from a user's xSides product.
- Automatically updates sides as newer versions become available.
- Allows sides to include dependent files.
- Enables a registered user to travel with their xSides configuration.
- Identifies sides installed by a user, the download source for these sides, and the use of those sides by a user.

AllSides Default Configuration

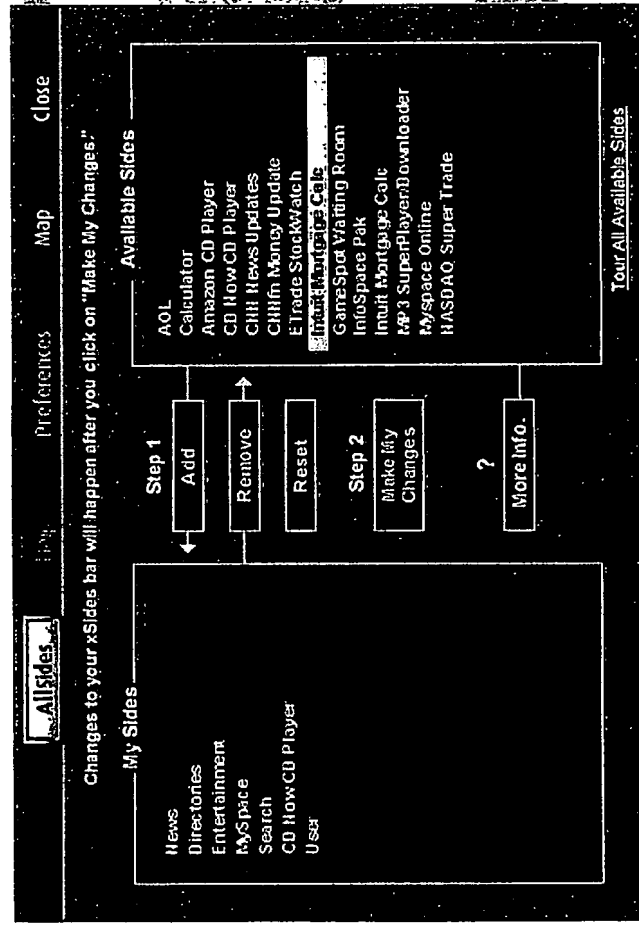
When a user downloads the xSides product for the first time, it includes a default configuration specific to the download source.

The configuration includes:

- Logo (appearing to the left of the bar)
- Base set of sides
- Class assignment
- GUID generation and server registration

AllSides Adding and Removing Sides (1)

The user can add and remove sides by selecting the AllSides dialog from the help menu.



AllSides Adding and Removing Sides (2)

- Available sides displayed in the right list box (based on user's class).
- Installed sides appear on the left.
- When the "Make My Changes" button is pressed, the xSides application will:
 - Update the local copy of the user's configuration file
 - Retrieve the side definition file(s) from the server
 - Post the updated user configuration file to the server
 - Regenerate xSides to display the selected changes

AllSides Future Enhancements

As a future enhancement, sides will be available for download directly from an HTML page.

- Every side that is added to a user's configuration includes a Distribution Vendor ID (DVID).
- DVID allows for flexibility in tracking users' activity on a side, based on the download source.

All Sides Side Installation

- Sides that are downloaded from the server are placed in a temporary folder on the local file system.
- A “package” is defined as a series of files required for a side to be successfully installed (dependent INI or DLL files, for instance).
- Sides are not available for installation until all elements of the package are available locally, and the application version meets the minimum version required by the side.
- Sides are installed by being “promoted” from the temporary folder into the working folder. Side promotion happens automatically when all criteria is met.

AllSides Traveling With xSides

If a user has registered, they can to download their configuration to any workstation running the xSides application by simply logging in using their username and password.

- User configuration file is retrieved from the server
- Any sides that do not already exist on that workstation are installed
- Feature provides a familiar, consistent interface with the application for those users that use multiple workstations frequently (for instance, both at work and at home)
- Changes made on either workstation are automatically synchronized with the server, even if both instances of the application are running simultaneously

AllSides Automatic Update

- Every side is described by a separate file, which includes an embedded date/time stamp.
- The xSides application periodically checks all the sides cached locally against the revisions available on the server.
- New side files are downloaded in the background and automatically installed.
- Sides that have been removed from the user configuration files of all users on the client will be deleted from the local file system.

User configuration files that have been unused for 30 days will also be removed from the local file system to conserve disk space, though the data is still retrievable from the server if needed after 30 days.

xSides File System Details

- Side files are stored in a separate folder from the user configuration files.
- Side files contain the data required to generate a side, a revision date/time stamp, dependent files (if any) and the minimum executable version (if required).
- DVIDs, sides, user buttons and the order in which all sides appear on xSides are all stored in the user configuration file, along with a revision date/time stamp (GMT).
- All files are stored encrypted.

Communications Layer (CommLayer) Overview

The CommLayer provides xSides dependencies conduit for communication with the xSides server. Each xSides user is identified by Global Unique Identifiers (GUID).

- Client/server architecture.
- XML driven.
- Efficient - all communication is coalesced to minimize Internet traffic.

CommLayer Features (1)

- Ping (scheduled server communication).
- Allows dependent components to register/unregister with CommLayer for ping callbacks.
- Schedules task for dependent components.
- Hot swapping of newly downloaded dependent components (excluding the CommLayer and the xSides executable).

Commlayer Features (2)

- Loading and unloading of dependent components (e.g., IA, Stats, mktplace) at the start and end of every xSides session.
- Auto updating of system files (happens in the background).
- Provides a conduit for communication between dependent components and xSides.
- Extensible.

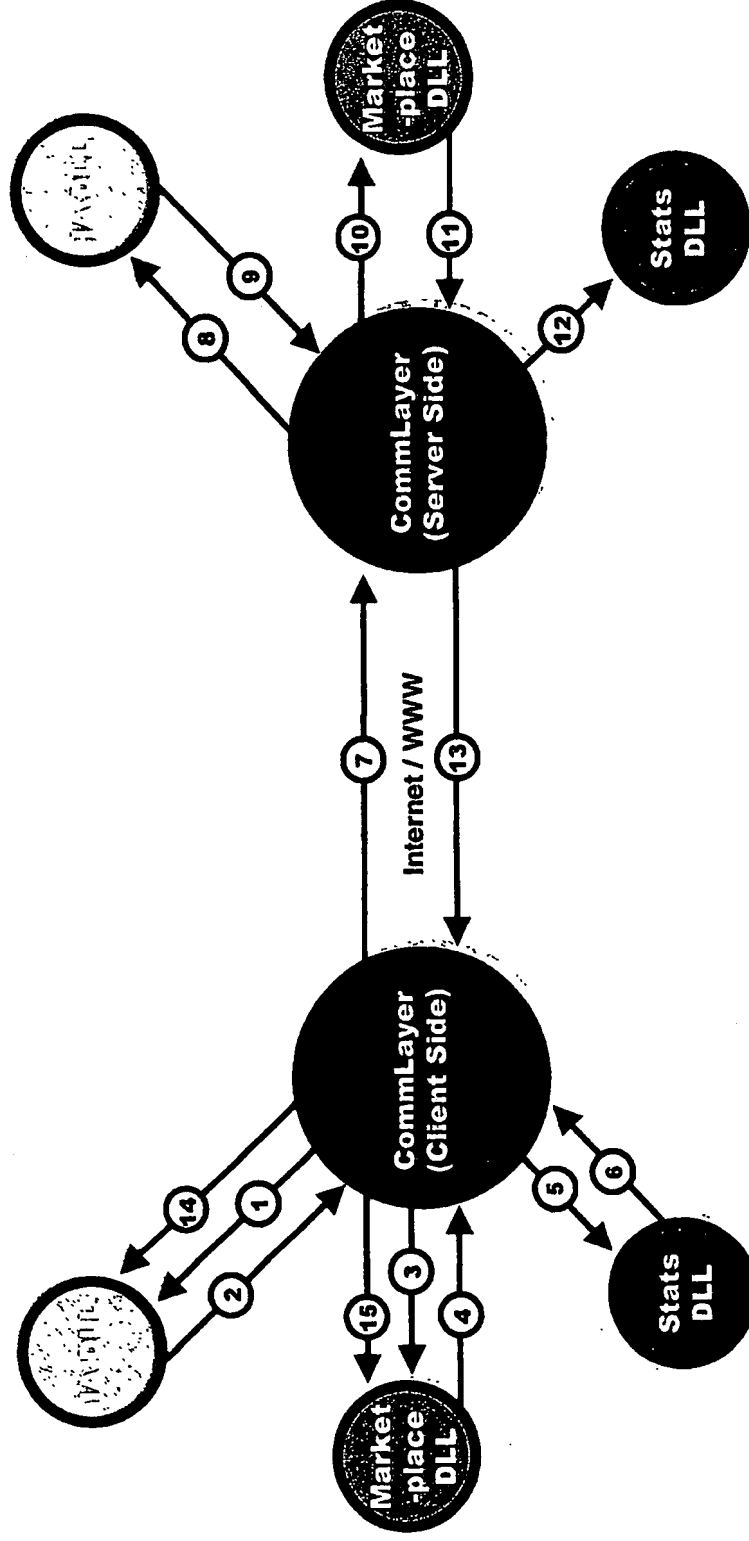
CommLayer Ping Functionality (1)

- Runs on a server-assigned time interval.
- Queries all of the registered DLLs and builds the needed markup which is then prepared to be sent to the xSides server.
- Encrypts the markup.
- Sends the markup to the server (GUID is sent with the ping to identify the user).

CommLayer Ping Functionality (2)

- Server-side CommLayer DLL parses and passes the markup to registered DLLs on server side.
- These registered DLLs process the markup and return their responses back to the server side CommLayer.
- Server encrypts the responses and sends them back to the client-side CommLayer.
- Client decrypts, parses, and passes responses back to appropriate registered DLLs.

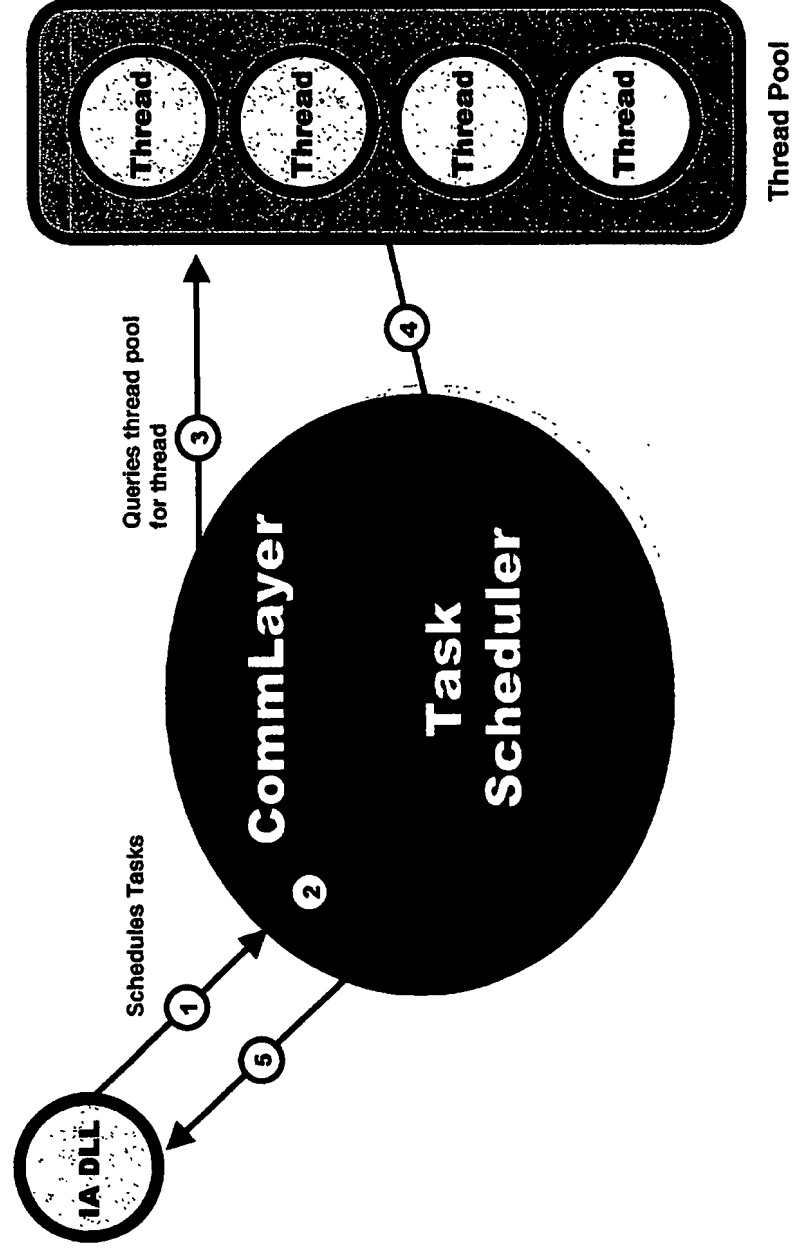
CommLayer Ping Functionality (3)



CommLayer Task Scheduling (1)

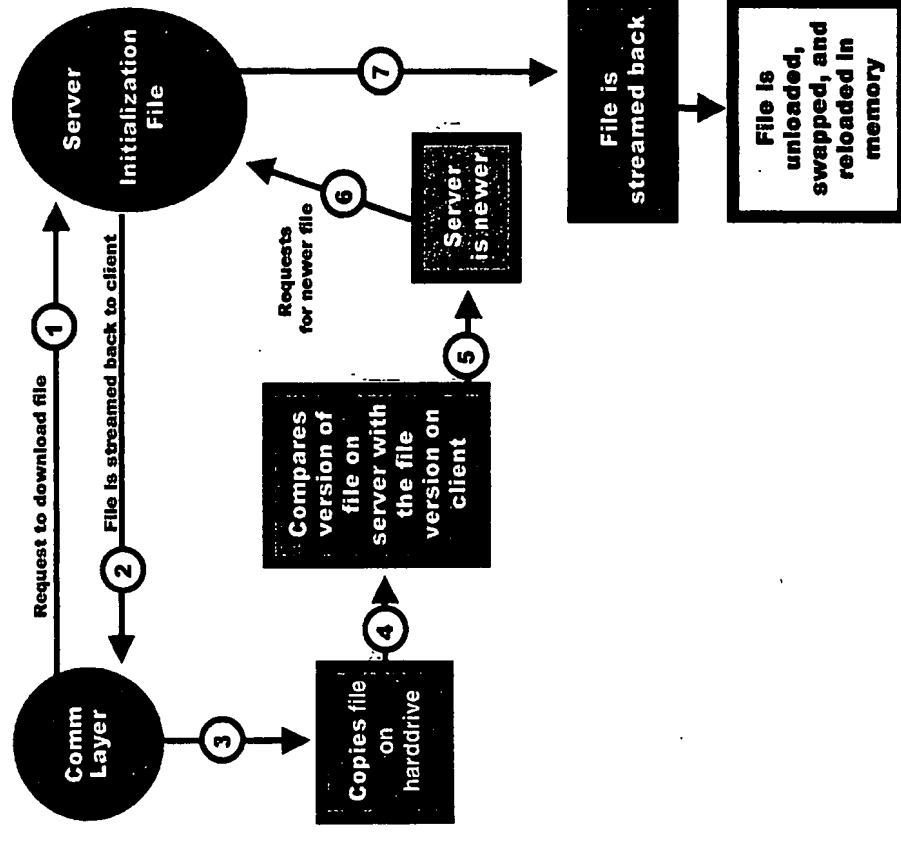
- **Manages the thread pool which avoids communication congestion and optimizes client performance**
- **Receives requests for tasks to be performed from the dependent DLLs (e.g., IA, Stats, etc.)**
- **Schedules and assigns tasks to the threads as threads become available in the thread pool**
- **“Do it now requests” - similar to task scheduling but provides threads immediately for tasks that cannot wait to be scheduled**
- **Governs overall number of threads that are made available depending upon the type of Internet connection**

CommLayer Task Scheduling (2)



CommLayer Hot Swapping

- Queries server for updated application components
- Downloads components as needed
- Registered DLLs are unloaded, swapped, and reloaded by the CommLayer and application files (e.g.: xSides executable) are queued for swapping on system startup



CommLayer Dynamic Configurability

Application variables are stored on the server and cached locally – this allows tuning for all clients.

Examples of these variables are:

- Ping interval (time interval between two pings)
- Connectivity time interval
- Ping URL
- Upload URL

CommLayer Extensibility

- CommLayer initialization file is downloaded from the server periodically.
- This file holds the information for loading/unloading of DLLs and executables.
- It also has server version of these DLLs.
- CommLayer compares this version with the client side version, and if the server has newer version it downloads it and hot swaps it with the older version of the DLL.
- By using hot swapping and dynamic configurability, xSides can introduce a new DLL with new functionality, that conforms to the rules of the CommLayer.

CommLayer Misc. Features

- Provides functions to upload and download files from the AllSides server (this functionality is used extensively by Marketplace DLL).
- This functionality can be extended to talk with any other server.

Instant Alert Overview

Instant Alerts provide a means for partners to send priority messages to their xSides users. Content is delivered to the client in standard HTML and displayed in a browser window. Users are identified by Global Unique Identifiers (GUID).

- Client/server architecture
- Markup driven using xSides communication layer
- Instant Alert content is HTML
- Server-side partner IP authentication
- API available to automate message generation

Instant Alert Features

Instant Alerts may be sent to single users or all users of an xSides side. Alerts may also be sent using templates, similar to a form letter, with or without replaceable parameters.

- Send Instant Alerts to a single user.
- Send Instant Alerts to all users of a side.
- Send templated Instant Alerts.
- Partner support for retrieving xSides user GUID.

Instantiate Alert Client Responsibility

- Schedules task thread with communication layer.
- Registers/unregisters for callback messages.
- Monitors for presence of URL in initialization file.
- Writes URL to initialization file.
- Responds to communications layer requests.

Instant Alert Data Flow

Below is a diagram of the sequence of requests and responses involved in retrieving an Instant Alert.

1 IA REQUEST

2 RESPONSE URL

3 URL REQUEST

4 INSTANT ALERT HTML

5 CONFIRMATION



Instant Alert Data Flow

Sequence of client requests and server responses:

- 1 Ping requests are made to the server
- 2 Response contains xSides markup which may contain a URL
- 3 If URL is present, it is written to an initialization file
- 4 xSides app monitors initialization file for URL
- 5 If URL is present, user is informed message waiting
- 6 User reads message by accessing Instant Alert URL
- 7 By closing Instant Alert, user accomplishes confirmation that the message was read

1 PING

2 RESPONSE URL

3 URL REQUEST

4 INSTANT ALERT HTML

5 CONFIRMATION



Instant Alert Templates

Instant alerts that are going to be used over and over with only minor changes to the content might be better suited to template style Instant Alerts. The template can be created once and used repeatedly by an xSides partner.

- Tools provided for creating and deleting templates.
- Tools provided for sending templated messages.
- Ability to create named attributes which get filled in when sent.

Instant Alert Partner Support

We provide a means for xSides partners to associate an xSides user's unique ID with the partner user's ID. The partner could create a button with a URL such as: <http://www.fabuloussite.com/scripts/Login.dll?Login&#pixid>

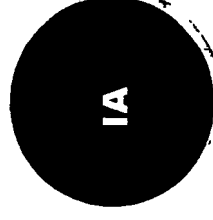
The “#pixid” is replaced with the user's GUID before it is exercised, thus providing the partner with the xSides user GUID. An xSides partner could then store this information with their own user ID. With this information in hand, the partner is able to send Instant Alerts to their xSides users.

- Partner user IDs can be associated with xSides user IDs.
- Macro replacement in URL provides access to xSides user GUID.

Instant Alert Client Class Diagram

The Instant Alert object derives from the communication layer, which manages server requests and delivers server responses.

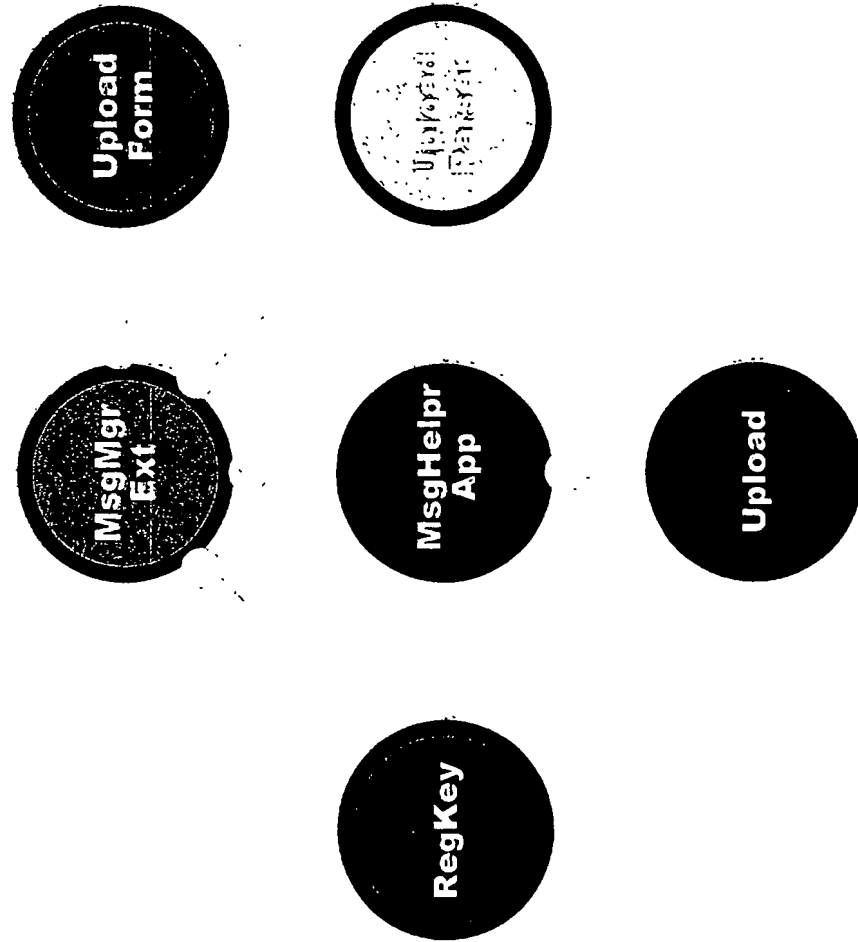
- Communication layer asks for requests from Instant Alert.
- Communication layer delivers responses via callback functions in Instant Alert.
- All client/server communication is through xSides markup.



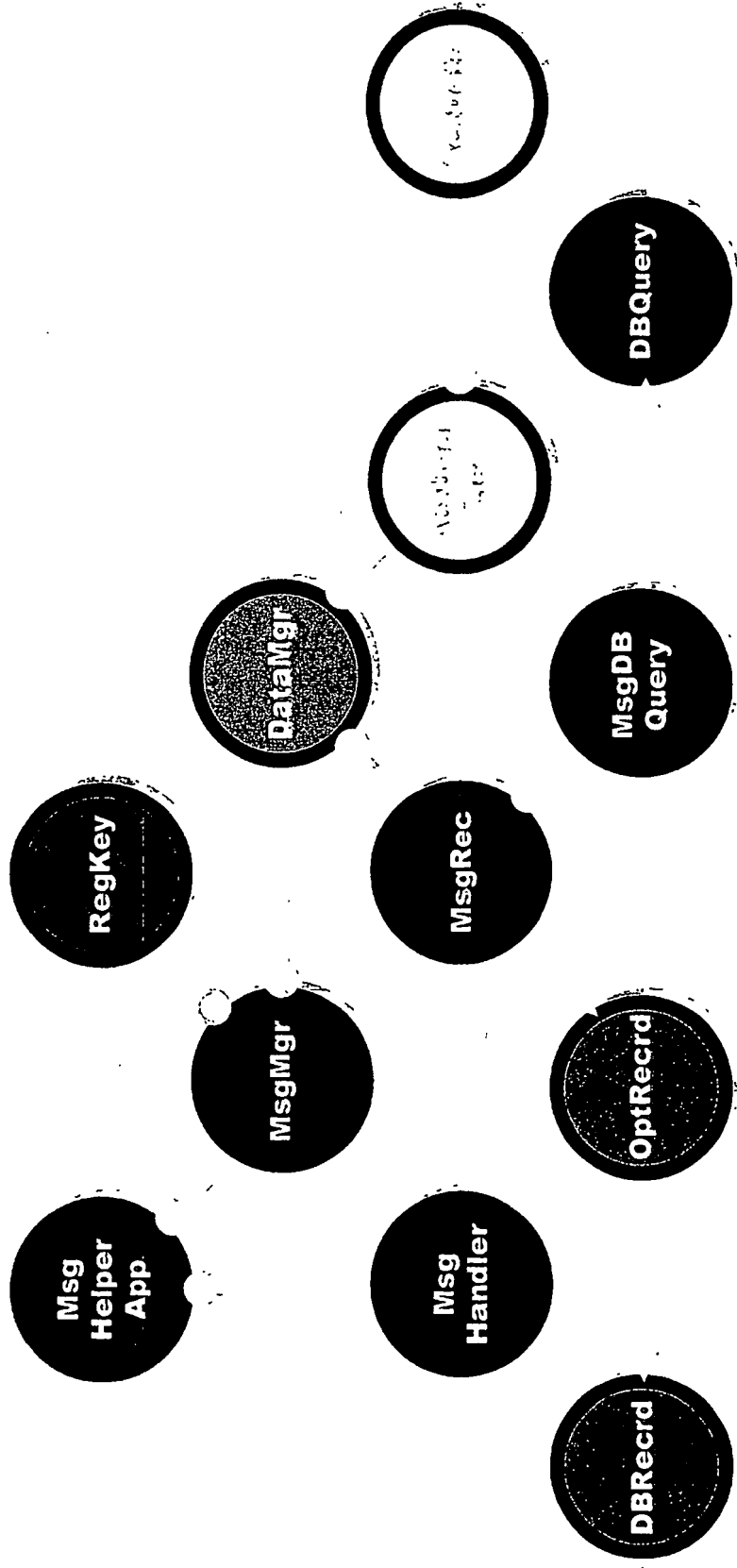
Instant Alert Server Responsibility

- **Delivers Instant Alert via HTTP request.**
- **Manages active user list.**
- **Provides IP security.**
- **Message delivery to and from database.**

Instant Alert Server Class Diagram A



Instant Alert Server Class Diagram B

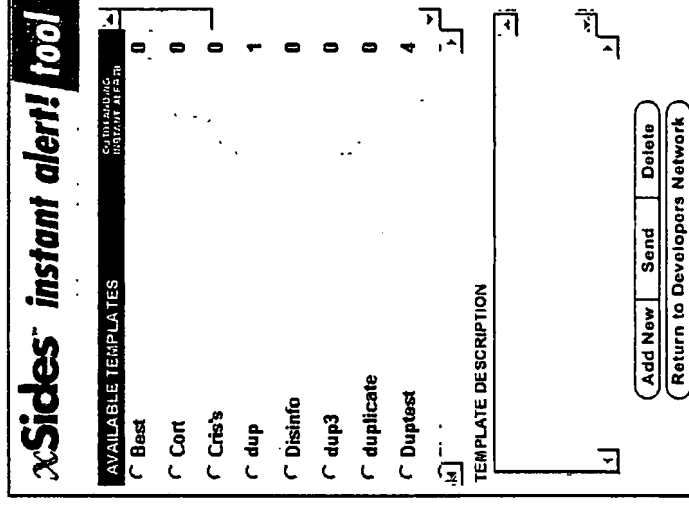


Instant Alert Template Tool Overview

Instant Alert Templates can be created, modified and sent, all from one location.

Templates are a way to reuse a generic message for more than one user, and more than one time, like a form letter.

- Create templates in HTML
- Save and send from the same tool
- Easy to use, Web-based interface



Instant Alert Tool Template Selection Page

Lists all the templates for your company, as well as outstanding alerts for each template.

- Outstanding Alerts can be looked up quickly and easily
- Partner-defined descriptions remind you of a template's content
- Server-side partner authentication

xSides
instant alert! fool

AVAILABLE TEMPLATES		OUTSTANDING INSTANT ALERTS
<input type="checkbox"/> Best		0
<input type="checkbox"/> Cont		0
<input type="checkbox"/> Crisis		0
<input type="checkbox"/> dup		1
<input type="checkbox"/> Disinfo		0
<input type="checkbox"/> dup3		0
<input type="checkbox"/> duplicate		0
<input type="checkbox"/> Duptest		4
<input type="checkbox"/>

TEMPLATE DESCRIPTION

Instant Alert Tool Add New

Create a new template quickly and easily, using HTML. Templates and their descriptions show up immediately on the selection page, once they have been saved. Send as soon as you're finished.

- Create templates in HTML
- Dynamically replaceable fields
- 2,000 character template size
- Save in our secure database with a single click

The screenshot shows a web-based form titled "xSides instant alert! tool" with a sub-header "ADD NEW TEMPLATE". The form contains two main input areas: "TEMPLATE NAME" and "TEMPLATE DESCRIPTION", both with text entry fields and small icons for undo and redo. Below these is a "CUT & PASTE CODE:" section with a text area containing the following HTML code:

```
<HTML><BODY>
<P> H1 <ARKTEXTNAME=FILENAME>
Send me<ARKTEXTAMOUNT=NUMBER>
This is a chain letter.
<BODY></HTML>
```

 At the bottom right of the form are three buttons: "Save", "Preview", and "Cancel".

Instant Alert Tool Send

Intuitive Web-based interface lets you replace dynamic fields easily, and preview what your user will see. Set the expiration date and send to either an individual user or all users of a certain side.

- Simple replacement of dynamic fields by entering information into form
- Preview results identical to user experience
- Send to either individual user or all users of a single side

The screenshot shows a web-based interface for sending alerts. At the top, it says "xSides instant alert! tool" and "SEND TEMPLATE". Below this is a "REPLACEABLE FIELDS" section with a "Show Preview" button. The main section is titled "SENDING OPTIONS" and has two tabs: "SINGLE USER" and "BROADCAST". The "SINGLE USER" tab is selected, showing a "Globally Unique ID (GUID)" field. The "BROADCAST" tab is also visible, showing "SIDE" and "#USERS" fields. Below these are "URL:" and "Display URL" buttons. The "Exp. Date:" field is set to "2000" and "Feb", and the "Exp. Time:" field is set to "12" and "00". At the bottom are "Send" and "Cancel" buttons.

Stats Overview

The Stats DLL is an object that resides on an end user's computer (the client) as a part of the xSides application.

When the xSides application is running on an end user's machine:

- Stats records the user's xSides activities.
- Recorded activity is sent to an xSides server and logged.
- Logged activity records can then be used for accounting purposes.

Stats Recorded Activities (1)

Examples of recorded activities are “clicks” and “impressions.”

- Click: a mouse click on a side or portal.
- Impression: the amount of time a given area of xSides is displayed to the client.
- Our Impression is a true impression. It is the length of time content is shown, i.e., true screen time.

e.g., if a user rotates the bar to Company A's side, then an impression will be the length of time this specific side is displayed. If the client clicks on this side, then that click is recorded.

Stats Recorded Activities (2)

The xSides executable tells the Stats DLL:

- What activity to record, (e.g., when the application displays an image on xSides, it tells the Stats DLL to start recording the impression time).
- When to record the activity (e.g., when the executable traps a mouse click, it then tells the Stats DLL to record the click).
- Where the activity happened: portal, or side.
- Which company's content the activity happened in.

Stats How It Works

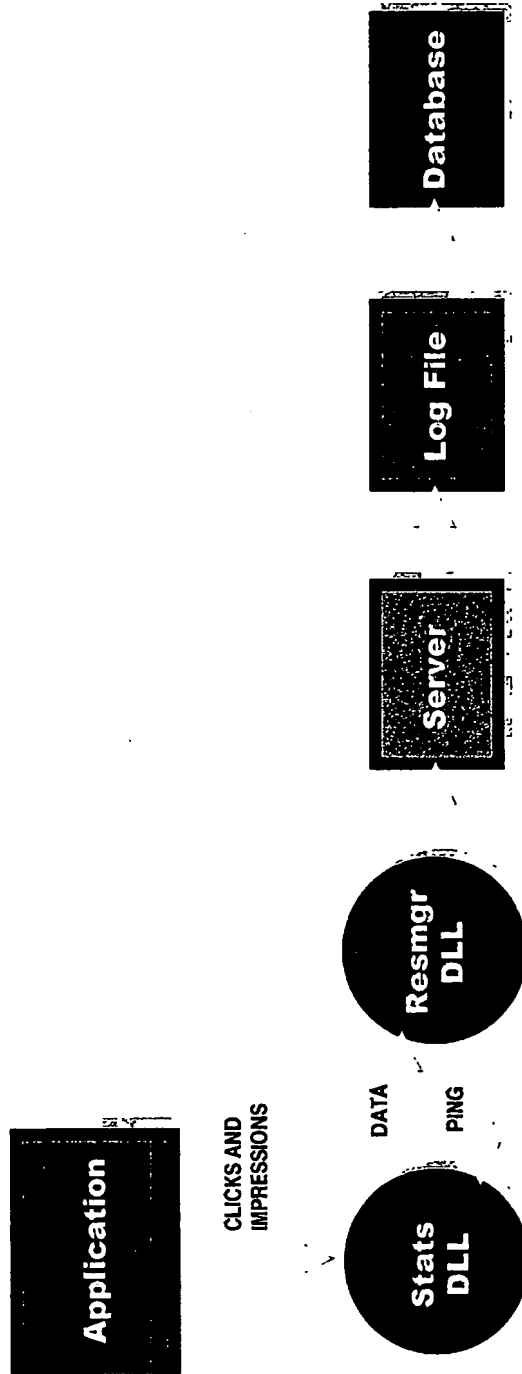
Application tells the Stats DLL to record an impression or click and passes the application name, the identifier of the content, its version, and the type of activity for recording to the Stats DLL.

1. A user clicks on a button in Company A's version 2.2 side.
2. The names –“side, Company A, 2.2, click” are recorded with the Stats component.
3. The number of clicks for “side, Company A, 2.2” is incremented by one.

Stats Information to the Server (1)

- 1. Resmgr DLL, another xSides DLL that resides on the client machine, pings Stats every minute at a scheduled interval.**
- 2. Stats sends a markup string to the Resmgr DLL containing its data for the previous time period.**
- 3. Resmgr DLL then uploads this to the server and the server logs this markup string into a log file for that day.**
- 4. The Logger DLL knows how to parse these strings and enters the data into a database.**
- 5. Invoices then can be generated from this database.**

Stats Information to the Server (2)



Stats Validity of Impression Time

Impression time for an xSides side begins when it is first displayed to the user, and ends when another display replaces it.

1. Timeout value is set when Stats is first loaded.
2. When Stats first gets pinged by Resmgr, it receives a timeout value from the server.
3. For each successive ping, idletime value is sent to Stats.
4. Idletime value is compared to the timeout value.
5. When idletime value is greater than timeout value, impression time is cut off.

Other situations are handled similarly, such as minimum impression time.

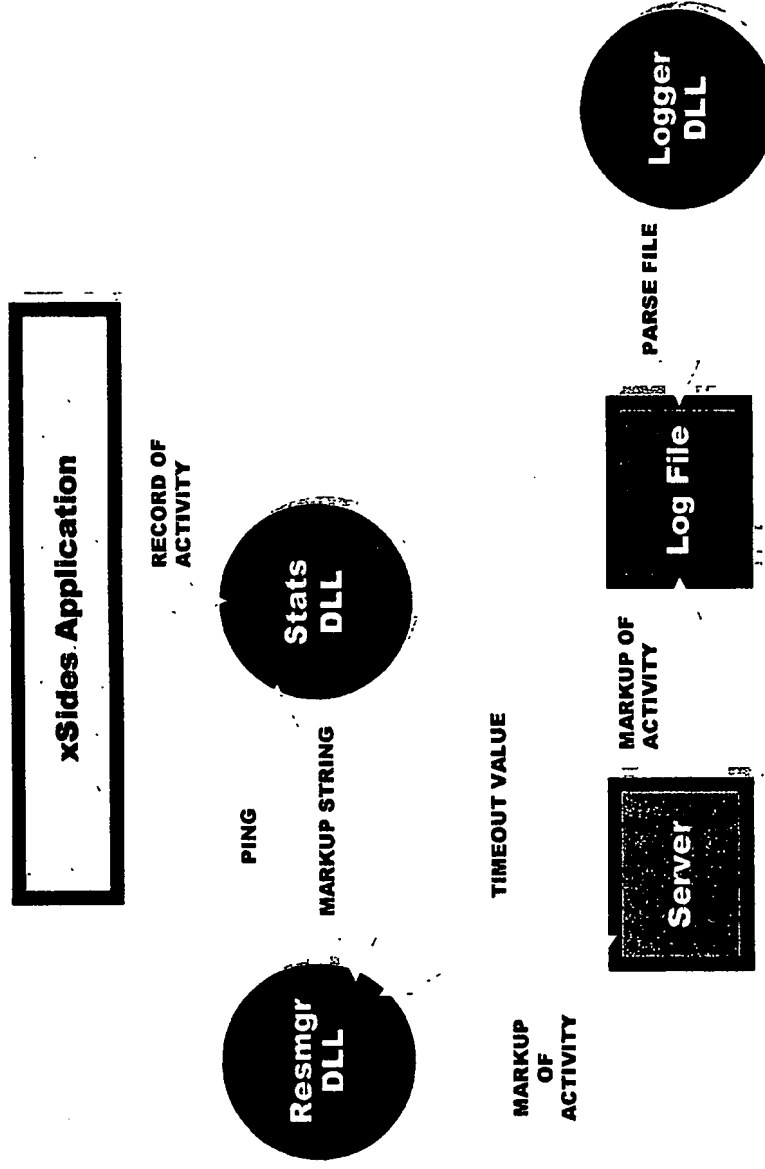
Stats Record of Activity

The data that is sent to the server to be logged is a string of characters that is in XML format:

```
<STATS><PORTAL><PIM Click=6 Impression=57 DVD=1.2></PIM><COMPANY A  
DVD=2.2></COMPANY A><COMPANY BDVD=2.3></COMPANY B></PORTAL><XSIDES><PIM  
DVD=4.0></PIM><COMPANY A Click=12 Impression=16 DVD=1.2></COMPANY  
A><COMPANY B Impression=34 DVD=1.2></COMPANY B></XSIDES></STATS>
```

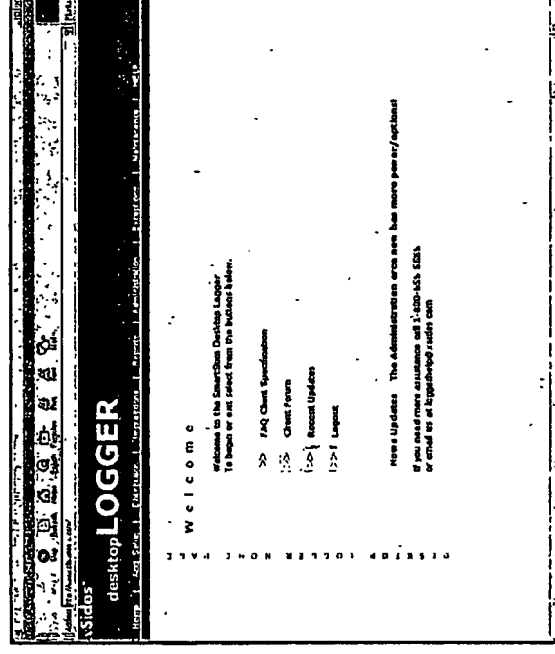
- String is downloaded to the server and logged.
- Logger processes the data for reporting and billing.

Stats Object Diagram



Desktop Logger Overview

- Desktop Logger calculates, stores, and retrieves activity-based statistics for a Web site or executable (xSides).
- Desktop Logger is used as an accounting tool that is rule-based: e.g., Desktop Logger uses pricing information to generate invoices and reports.



Desktop Logger Tracking

Web site activities that Desktop Logger tracks:

- Clicks
- Users (Through an anonymous ID)
- Visits (Defined by time intervals)
- Unique visits (The first visit only)
- Destination URLs
- Source location (xSides side)
- Version/distribution information

Desktop Logger How It Works (1)

Step 1

Two methods send markup to our xSides servers:

- Redirected links
- “Ping” from an application

What is markup?

Markup allows us to send detailed information to our servers that would not normally be sent.

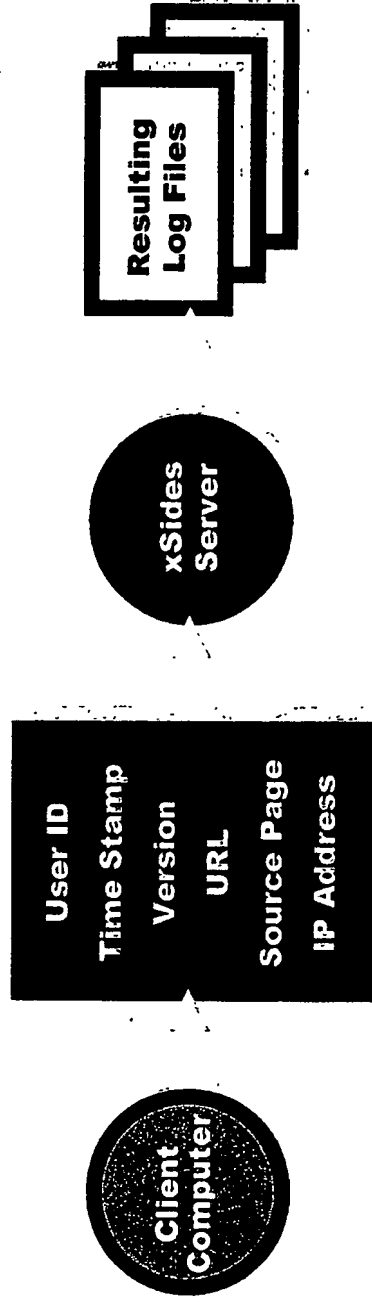
Sample Decrypted Markup:

```
<REQUEST%11GUID=0M555A6180-B6EC-11D3-BD0F-00902716AAB6%11SIGNON=1%09APPVER=PXGEN><PARAM%09AC=GET></PARAM><INI_VER%09AC=GETLIST></INI_VER><STATS%09AC=GET></STATS><MKT%09AC=SIGNON></MKT><MKT%11AC=GETSIDELEFT></MKT></REQUEST>
```

The markup goes into the natural storage mechanism of the Web server so it can't be adulterated.

Desktop Logger How It Works (2)

Possible Data Transferred Through Markup



Desktop Logger How It Works (3)

Step 2

- Desktop Logger parses the Internet logs.
- Compresses the pertinent data into a proprietary format.
- Inputs data into the back-end SQL database.

Step 3

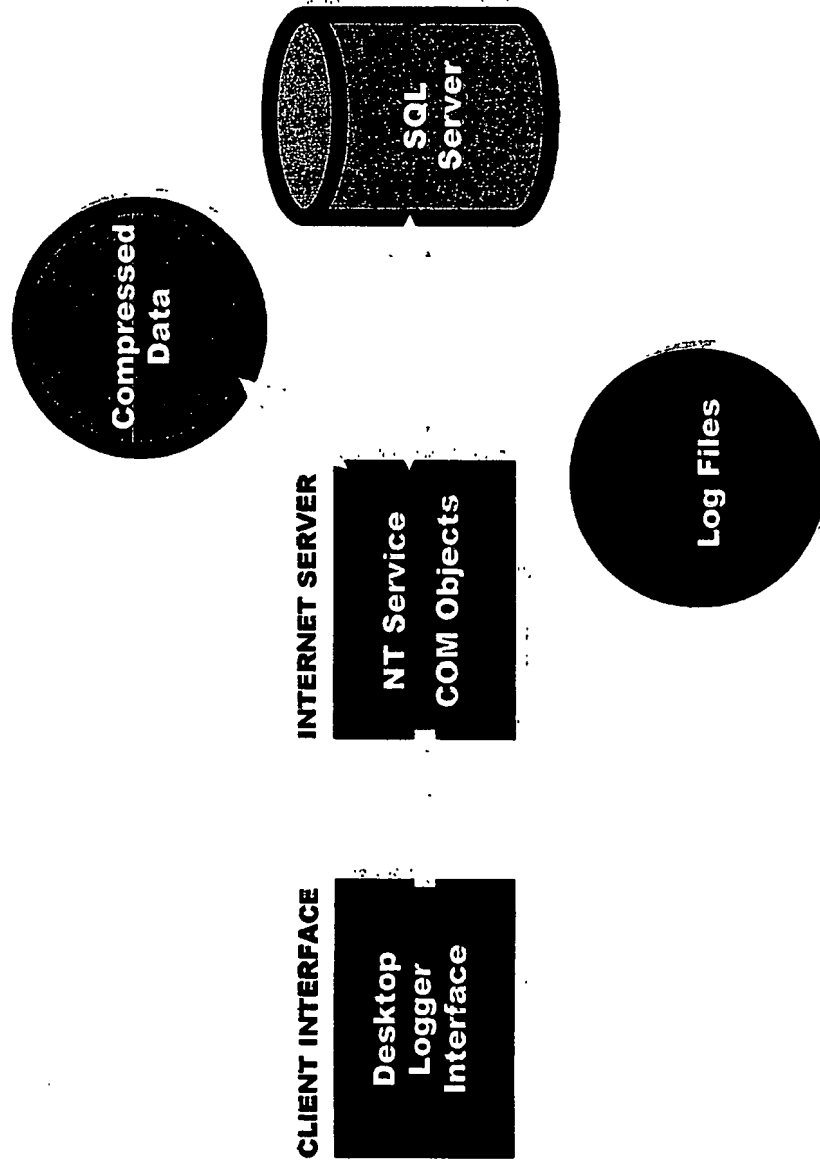
The Desktop Logger interface accesses the SQL database and compressed files in order to generate reports and invoices.

Desktop Logger 3-Tiered Architecture

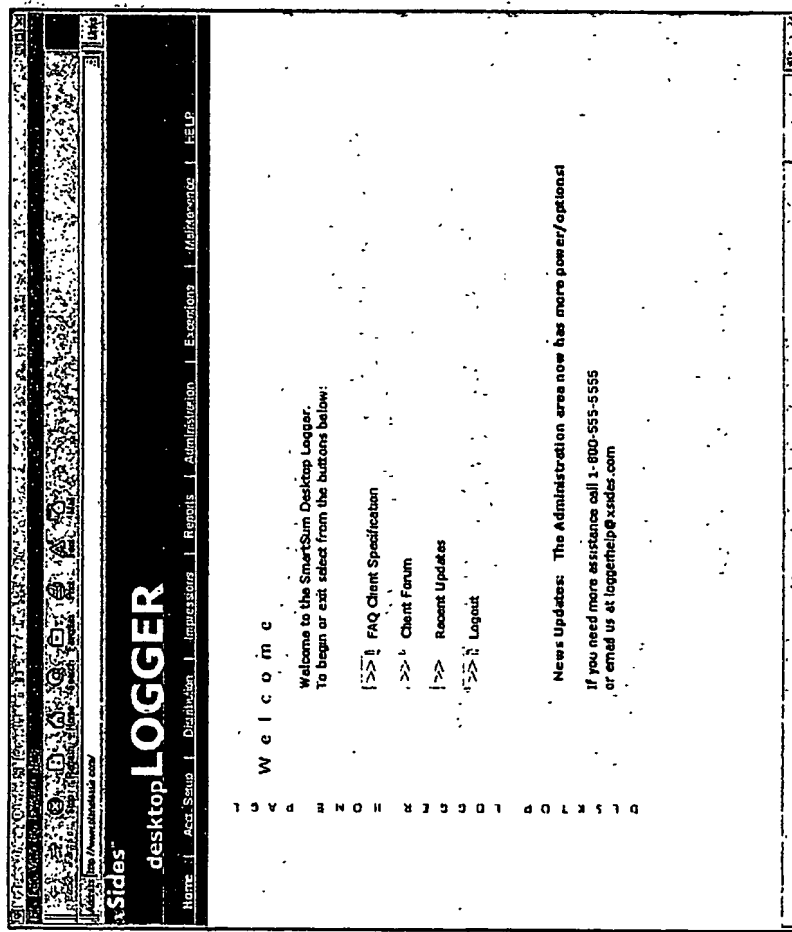
Desktop Logger is an application that consists of 3 main parts:

- The front-end is a Web-based client interface called Desktop Logger.
- Middle tier processes raw IIS server logs – the work horse of Desktop Logger.
- Backend consists of a large SQL database that stores for partner, distributor and statistical information.

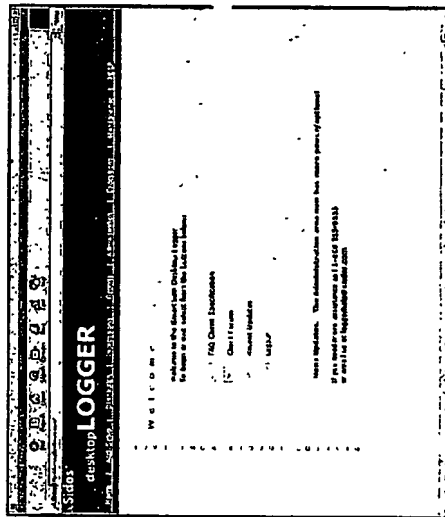
Desktop Logger 3-Tiered Architecture



Desktop Logger Front-End



Desktop Logger Front-End



Desktop Logger Middle Tier

- Processes raw log files automatically from numerous servers.
- Compresses data into a proprietary format.
- Automatically inserts data into SQL database.
- Supports ActiveX objects that are accessible through a Web interface to generate reports and invoices.



Desktop Logger Third Tier

SQL database contains:

- Tables
- Stored procedures
- Triggers

SQL database stores:

- Invoicing statistics
- Pricing rules
- Operator logins and security levels
- Company information
- Version, source, URL information
- Exceptions

